

# Platform Instrument Hierarchy -- ECSE-90

Author(s): <Dana Shum>

## Description of Problem

There are 13 issues related to the structures and relationships surrounding platforms, instruments, and sensors. The discussions around these issues are focused in two main areas, 1) the use of "Sensor" and 2) the need for a consistent and formal definition of the platform/instrument relationships.

### JIRA Linkage

 [ECSE-90](#) - JIRA project doesn't exist or you don't have permission to view it.

<Ctrl+Shift+J allows you to link to your ECSE JIRA ticket easily. Replace the ticket above with your ticket>

## Background

### Instrument vs Sensor

The use of the terms "Instrument" and "Sensor" has been a source of confusion and debate throughout the evolution of ESDIS metadata standards development. GCMD's DIF standard used `SensorName` to represent an instrument. ECHO-10 later attempted to model the complexity of modern instruments by creating an instrument object that can be composed of multiple sensors. This allowed for a more detailed description of instruments like the [SMAP](#) instrument that is composed of an L-band radar and an L-band radiometer. In SMAP, the data of the radar and radiometer are combined to produce soil moisture measurements. Debate continues to this date over the semantics of instrument versus sensor.

As ISO-19115 was developed the decision was made to eliminate the use of the term "sensor" and the ISO version of Acquisition Information only includes an instrument (`MI_Instrument`) class. Recently though, the debate has been re-opened in ISO with a proposal to add an `MI_Sensor` class to ISO as a sub-type of `MI_Instrument`. It is interesting to note that the `MI_Instrument` and `MI_Sensor` classes are identical in their attributes.

The only difference between the two classes is that the `MI_Instrument` has a "contains" relationship to the `MI_Sensor`. It appears that there is little semantic difference between the objects. This difference lies in the fact that multiple instruments can be combined to produce a new instrument. If the `MI_Sensor` class is removed and the relationships between instruments are retained then a rich structure for describing the instruments provided on a modern platform can be expressed without the need for a semantically void class representing a "Sensor". This can provide a simple resolution that eliminates the Sensor/Instrument debate for the entire CMR community.

### Platform Instrument Hierarchy

Describing the hierarchy and relationships for platforms and their instruments is another area that has been a source of debate as evidenced by the discussions surrounding Acquisition Information in recent UMM reviews. DIF representations of Platform/Instrument were focused upon a keyword taxonomy that addresses the breadth of instrument types as well as the diverse nature of platform types (see [GCMD Keyword Definitions](#)). As discussed above, ECHO created a simple hierarchical definition of Platform that contained multiple Instruments along with key characteristics of each. Both ECHO and ISO have links to the DIF keywords as part of their Platform/Instrument hierarchies. This approach retains the elegant taxonomy of the DIF with a richer structure for describing the Platform and Instrument. Both approaches suffer from the lack of an authoritative source for the relationships between Platforms and Instruments. We currently have 9 open issues related to the Platform/Instrument hierarchy. The discussion around these issues clearly presents the advantages and disadvantages of each of these approaches and demonstrates that a common agreement on an approach that leverages the best features for the UMM community is required.

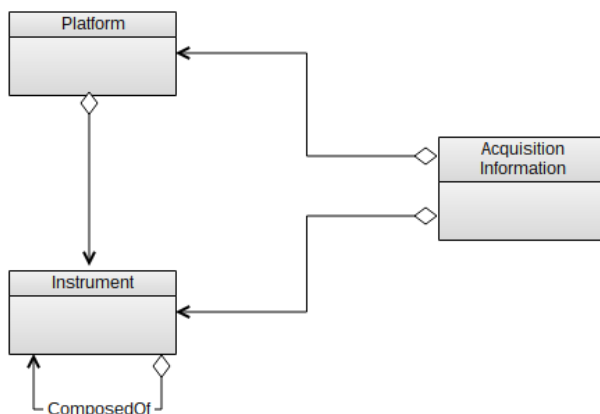
## Approach

For UMM and CMR, we propose the following fundamental tenets to guide the resolution of issues related to the Platform/Instrument hierarchy:

- *Focus on the CMR community and their metadata uses.* Metadata standards, recommendations, and guidance only make sense in the context of a community and their use cases for the metadata. When deciding between options for a UMM metadata structure, we choose the one that makes sense for the CMR community. We learn from other communities, such as the ISO community, but our focus is on the metadata needs of the CMR community.
- *Learn from ISO, but don't blindly reuse ISO metadata structures.* Whenever possible, have a single UMM metadata structure that might map to multiple locations in ISO. We don't need the flexibility and complexity of ISO to support the rich metadata needs of the CMR.
- *Rationalize the use of inconsistent metadata definitions used by other standards to produce an efficient and systematic definition that meets current and future needs of the CMR community.*

## Recommendations

**Recommendation 1 – Eliminate the use of a "Sensor" object in UMM.** Instead, use the ISO proposal that allows an instrument to be composed of one or more other instruments, but do not include the MI\_Sensor subtype of Instrument. The idea of "sensor" will be implemented using instrument composition, and a mapping to DIF sensor object will be provided as an example. Figure 1 shows the relationships between the Acquisition Information, Platform and Instrument objects. Note that the Instrument object now has a recursive relationship that is titled "ComposedOf". This relationship enables a hierarchy of Instrument objects where each instrument can be composed of one or more other instruments. This provides the capability to model complex instruments without resort to a Sensor object. For reference purposes, you can see the ISO version of these objects [here](#).



**Figure 1 – Relationships between Platform, Instrument, and Acquisition Information.**

Affected issues include:

**ECSE-71** – Eliminate sensor as described above, but maintain separation of Platform and Instrument. A UI may choose to present only the Platform-Instrument combination for their users, but we don't want to lose the instrument mapping information in UMM.

**UMMC-429 / UMMC-167 / EDSC-698** – Eliminate sensor as described above.

**Recommendation 2 – CMR should be the authoritative source for Platform/Instrument relationships for all NASA metadata and should follow the structure shown in Figure 1 above.** This figure shows the intrinsic relationship between Platform and Instrument that allows UMM to support the controlled hierarchy of Platforms and Instruments. By incorporating these relationships in UMM we define and enforce the hierarchical controlled structure, including the links to specific platform & instrument keywords. Both the Platform and Instrument object retain their references to specific GCMD Keywords, but we recommend including standard keywords for "Not Available", "Not Applicable", and "Unknown". Adopt the UMM governance model for modifying and extending the hierarchy.

Note that the CMR may need to support the controlled ingest of Platform/Instrument relationships from authoritative sources outside of NASA. A new issue will be created to support this activity.

Affected issues include:

CMR-1743 / CMR-1938 / UMMC-178 / MMT-394 / UMMC-203 / UMMC-134– Use the new platform/instrument hierarchy of UMM.

ECSE-15 - Also incorporate multiple mappings of UMM structures when mapping to ISO.

ECSE-101 - Was waiting for completion of ECSE-90, can now proceed.

ECSE-100 - Recommends normalization of Platform/Instrument data. This normalization is one of the requisites for establishing a controlled authoritative source for Platform/Instrument relationships. As such, it is subsumed by this issue. Recommend closing ECSE-100 in favor of ECSE-90.

## Changes to UMM-C fields

Remove Sensor from Acquisition Information, move the DIF Sensor mappings into the Instrument Section

Changes to UMM-Common fields

1. Eliminate the "Sensor" object from UMM-Common spec.
2. Add a recursive relationship to the Instrument object in UMM-Common to reflect the composition of an instrument. For example:
  - a. Platform/Instrument/ComposedOf (0..1)
  - b. Platform/Instrument/ComposedOf/InstrumentId (0..\*) { Reference to another Instrument Object serving as a Sensor for this instrument}
3. Remove the Instrument object from the Platform object. Add references from Platform to Instrument in the following fashion:

Platform/Instruments (0..1)

Platform/Instruments/InstrumentId (0..\*)

## Mappings to DIF, ECHO, ISO

1. Change UMM Instrument mappings to include the mappings to the DIF-10 Sensor class.

## Interoperability Considerations

We are not adopting or removing any classes defined by ISO or other standards. The changes recommended in this page will all map cleanly to all of the CMR supported dialects including ISO, ECHO, DIF, & DIF-10. As described above, some changes are required to the mappings but no information will be lost when mapping between UMM and these dialects.

## Changes to CMR

1. CMR needs to support the controlled hierarchy of Platform/Instrument and its relationships for NASA metadata and provide the ability to import Platform/Instrument hierarchies from other organizations.
2. Accommodate the removal of Sensor and the new mappings to ECHO (and DIF-10 if those mappings are required).

## What should the MMT forms look like

The MMT forms will need to be modified to support 2 use cases:

1. Editing a UMM Collection record – The existing forms that allow the user to fill in the instrument information, characteristics, etc. should be replaced with a simple mechanism to select the appropriate Instrument(s) from the CMR's controlled Platform/Instrument hierarchy. At that point, the Collection record will incorporate the information from the hierarchy, instead of creating its own version of the metadata.

2. Editing the CMR's controlled Platform/Instrument hierarchy. This use case is not available to the general metadata user. Instead, this is the primary mechanism for creating and updating values in the Platform/Instrument hierarchy.

## How should the values in these fields be presented to the user on the EDSC

There are likely no changes to the presentation of the Acquisition Information in the EDSC as it exists today. However, an issue should be opened for the EDSC team to evaluate the value of adding the recursive Instrument relationship.

## Reconciliation Issues

A reconciliation activity will be required to populate the Platform/Instrument structure. Currently the platform and instrument relationships are uncontrolled resulting in consistency problems across the NASA collections. We see multiple spellings, abbreviations, etc. for the NASA collections. As part of the migration to a controlled Platform/Instrument hierarchy, these consistency issues need to be reconciled and incorporated into the CMR's controlled hierarchy. To understand the scope of the reconciliation effort, a preliminary analysis was conducted on all of the collections in the CMR. A short summary of the results include:

- There are 149 unique platform names in the CMR which do not exist in the GCMD Platform Keyword list. Many of these are simple to fix typos such as "Weather Balloon" with 1 space versus "Weather Balloon" with 2 spaces, or "AERIAL PHOTOGRAPH " with a space on the end. Other items may need to be added to the GCMD Keyword List, or may be changed to comply with the existing keyword hierarchy.
- There are 255 unique instrument names in the CMR which do not exist in the GCMD Instrument Keyword list. As with the platform names, many of these are inconsistency problems, like:
  1. Moderate-Resolution Imaging Spectroradiometer
  2. MODIS (MODERATE-RESOLUTION IMAGING SPECTRORADIOMETER
  3. MODIS (MODERATE-RESOLUTION IMAGING SPECTRORADIOMETER)
- There are 481 unique sensor names in the CMR which do not exist in the GCMD Instrument Keyword list. Many of these are the same inconsistent naming problems as in the previous items and are straightforward to correct. However, there are a large number of these items that have names that are specific to a single international collection. *Further analysis is required to determine the correct handling for these items.*

A full list of the Platform, Instrument, and Sensor information can be found in [20160524 CMR PlatInstSensor Data.xlsx](#).